AF

Docket No.: 95-472 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re Application of	:	
DESAI	:	
Serial No.: 09/986,967	:	Group Art Unit: 2151
Filed: November 13, 2001	:	Examiner: TRAN, Nghi V.

For: ARRANGEMENT FOR PROVIDING CONTENT OPERATION IDENTIFIERS WITH A SPECIFIED HTTP OBJECT FOR ACCELERATION OF RELEVANT CONTENT OPERATIONS

MAIL STOP: APPEAL BRIEF – PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDED APPEAL BRIEF

Sir:

In response to the Notification of Non-Compliant Appeal Brief mailed August 14, 2007, the following Amended Appeal Brief is submitted.

This is an appeal from the final rejection of claims 1-4, 6, 8-9, 11, 13-16, 18, 20-23, 25, 27, 28, 30, 32-35 and 37 in the above-identified patent application.

This Appeal Brief is submitted as required by 37 C.F.R. §41.37.

1. Real Party in Interest:

This application is assigned to Cisco Technology, Inc., the real party of interest.

2. Related Appeals and Interferences:

There are no other appeals or interferences known to Appellant that will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

3. Status of Claims:

Claims 1-38 are pending in this application. Claims 1-4, 6, 8-9, 11, 13-16, 18, 20-23, 25, 27, 28, 30, 32-35 and 37 stand rejected by the Examiner. Claims 5, 7, 10, 12, 17, 19, 24, 26, 29, 31, 36, and 38 stand objected to by the Examiner as depending from a rejected claim, but reciting allowable subject matter. Claims 1-4, 6, 8-9, 11, 13-16, 18, 20-23, 25, 27, 28, 30, 32-35 and 37 are appealed.

4. Status of any Amendment File Subsequent to Final Rejection:

An Amendment After Final was filed on March 12, 2007 to correct informalities in claims 11, 25 and 36. The Advisory Action mailed April 3, 2007 did not specify whether the Amendment After Final would be entered for purposes of appeal, and a message left with the Examiner on July 10, 2007 has not been returned to date. It will be assumed herein that the Amendment After Final will be entered for purposes of appeal unless indicated otherwise by the Examiner, as the amendments do not raise any new issues but rather simplify issues on appeal: these amendments are reflected in the Claims Appendix *infra* for the claims on appeal.

5. Summary of Claimed Subject Matter:

The claimed subject matter includes independent claims 1, 13, 20 and 32, and dependent claims 2-12, 14-19, 21-31, and 33-38. Independent claims 1, 13, 20 and 32 each specify transfer of an HTTP response (e.g., 60 of Fig. 3B transferred by server process 32 of Fig. 1 in step 84 of Fig. 4A, or by proxy agent 22 of Fig. 1 in step 98 of Fig. 4B) to an HTTP request (HTTP Get Request in step 84 of Fig. 4A, or step 90 of Fig. 4B), where the HTTP response includes a first content object (34 of Figs. 1, 3A and 3B), having been requested in the HTTP request and a content operation identifier (e.g., 40 of Fig. 2A, 46a of Fig. 2B): the content operation identifier specifies a directive (e.g., 42a of Fig. 2A, 50 of Fig. 2B) for prefetching an identified second

content object (e.g., 44a of Fig. 2A, 52 of Fig. 2B), where the prefetching is a content operation that is distinct from presentation of the first content object.

Hence, any device (e.g., 16a or 16b of Fig. 1) receiving the HTTP response (e.g., 60 of Fig. 3B) is able to present not only the first content object (e.g., 34 of Fig. 3B), but also is able to ***prefetch*** (102 or 106 of Fig. 4B) the second content object (e.g., 44a, 44b, 44c of Fig. 2A, 52 of Fig. 2B) based on receipt of the content operation identifier (40 of Fig. 2A, 46 of Fig. 2B). Consequently, any device that receives the HTTP response can prefetch the second content object for acceleration of web content for a user.

The claimed subject matter addresses the problem in existing proxy cache techniques of requiring a request for the web content to have been previously requested by a client device, where a client device cannot enjoy any benefits of proxy caching if a prior client device has not previously requested the same web content (see, e.g., page 1, line 23 to page 3, line 2), without the necessity of additional resources executed by the client device (page 2, 4-20).

Hence, independent claim 1 specifies a method of providing content to a device (e.g., 16a of Fig. 1) according to Hypertext Transport Protocol (HTTP), the method comprising receiving an HTTP request (HTTP interface 20 of server 12b receives HTTP Get request in 74 of Fig. 4A; page 6, lines 24-26, page 9, lines 14-15) for a first content object (34 of Figs. 1, 3A, 3B, page 6, line 27 to page 7, line 7). The method also includes identifying (70 of Fig. 4A, page 9, lines 6-13) a content operation identifier (tag file 36 stores content operation tags 40 of Fig. 2A or extensible HTTP headers 46 of Fig. 2B, page 7, lines 11-15, page 8, lines 10-13) that identifies a corresponding second content object (44 of Fig. 2A or 52 of Fig. 2B, page 7, lines 16-18 and 21-24; page 8, lines 11-13 and 16-18, page 10, lines 10-11) determined as relevant to the first content object by a predictive caching operation (70 of Fig. 4A, page 2, lines 5-6 and page 9, lines 6-10), the content operation identifier including a directive (e.g., 42a of Fig. 2A, page 7, lines 16-17; 50 of Fig. 2B, page 8, lines 11-18) for prefetching (page 7, lines 12-24) the second content object as a content operation distinct from presentation of the first content object by the device (page 8, lines 7-9). The method also includes sending to the device an HTTP response to the HTTP request (84 of Fig. 4A, page 9, lines 19-25, page 10, lines 2-3), the HTTP response

including the first content object and the content operation identifier, enabling the device (e.g., 16b of Fig. 1) to perform the prefetching of the second content object based on receipt of the content operation identifier and distinct from the presentation of the first content object (steps 94 through 106 of Fig. 4B, page 9, line 26 to page 10, line 23).

Claim 2 adds to the method of claim 1, wherein the identifying step includes retrieving (76 of Fig. 4A, page 9, lines 14-18), based on retrieval of a first stored file (34 of Fig. 1) containing the first content object, a second stored file (36 of Fig. 1) associated with the first stored file and containing the content operation identifier.

Claim 3 adds to the method of claim 2, wherein the sending step includes adding to the first content object (34 of Figs. 1, 3A and 3B) a content operation tag (36a of Fig. 2A includes content operation tag 40 of Fig. 2A, page 7, lines 11-15) that specifies the content operation identifier including a directive tag (42a of Fig. 2A, page 7, lines 16-20) specifying the corresponding content operation to be performed by the device and an object identifier (e.g., 44a, 44b, 44c of Fig. 2A, page 7, lines 21-22) that specifies a location of the second content object.

Claim 4 adds to the method of claim 3, wherein the first content object is a Hypertext Markup Language (HTML) document (34 of Fig. 3A), the adding step including inline prepending (82 of Fig. 4A, 60 of Fig. 3B, page 9, lines 19-23) the content operation tag from the second stored file into the HTML document.

Claim 6 adds to the method of claim 2, wherein the sending step includes inserting into the HTTP response at least one extensible HTTP header (e.g., 46a of Fig. 2B, step 80 of Fig. 4A, page 8, lines 10-20; page 9, lines 19-20) that specifies the content operation identifier including said directive (50 of Fig. 2B) to be performed by the device and an object identifier (52 of Fig. 2B) that specifies a location of the second content object.

Claim 8 adds to the method of claim 1, wherein the sending step includes adding to the first content object (34 of Figs. 1, 3A and 3B) a content operation tag (36a of Fig. 2A includes content operation tag 40 of Fig. 2A, page 7, lines 11-15) that specifies the content operation identifier including a directive tag (42a of Fig. 2A, page 7, lines 16-20) specifying the

corresponding content operation to be performed by the device and an object identifier (e.g., 44a, 44b, 44c of Fig. 2A, page 7, lines 21-22) that specifies a location of the second content object.

Claim 9 adds to the method of claim 8, wherein the first content object is a Hypertext Markup Language (HTML) document (34 of Fig. 3A), the adding step including inline prepending (82 of Fig. 4A, 60 of Fig. 3B, page 9, lines 19-23) the content operation tag into the HTML document.

Claim 11 adds to the method of claim 1, wherein the sending step includes inserting into the HTTP response at least one extensible HTTP header (e.g., 46a of Fig. 2B, step 80 of Fig. 4A, page 8, lines 10-20; page 9, lines 19-20) that specifies the content operation identifier including the directive (50 of Fig. 2B) to be performed by the device and an object identifier (52 of Fig. 2B) that specifies a location of the second content object.

Independent 13 specifies a method of retrieving content for a device (e.g., 16b or 14b) according to Hypertext Transport Protocol. The method comprises first sending an HTTP request (resource 22 of proxy device 16a in Fig. 1 forwards request in step 92 of Fig. 4B, page 9, line 26 to page 10, line 2) for a first content object (34 of Figs. 1, 3A, 3B), received from the device (14b or 16b of Fig. 1, page 9, line 26 to page 10, line 2), to a destination server (12b of Fig. 1, page 10, line 2) specified by the HTTP request. The method also includes receiving from the destination server an HTTP response to the HTTP request (step 94 of Fig. 4B, page 10, lines 2-4) that includes the first content object (34 of Figs. 1, 3A, 3B) and a content operation identifier (40 of Fig. 2A or extensible HTTP headers 46 of Fig. 2B, page 7, lines 11-15, page 8, lines 10-13 and 21-26) that specifies a directive (42a of Fig. 2A, page 7, lines 16-17; 50 of Fig. 2B, page 8, lines 11-18 and 22-26) for prefetching an identified second content object (page 7, lines 12-24) as an operation to be performed on the identified second content object and distinct from presentation of the first content object. The method also includes second sending the first content object (98 of Fig. 4B, page 10, lines 5-6) to the device (14b or 16b of Fig. 1, page 10, lines 5-6). The method also includes executing the operation of prefetching the second content object in response to the content operation identifier (steps 102 or 106 by proxy agent 22 of proxy 16a of Fig. 1, page 10, lines 7-23).

Claim 14 adds to the method of claim 13, wherein the executing step includes detecting the content operation identifier based on parsing the HTTP response (step 96 of Fig. 4B, page 10, lines 2-4), and accessing the identified second content object for execution of the operation (steps 102 or 104 of Fig. 4B, page 10, lines 7-23).

Claim 15 adds to the method of claim 14, wherein the detecting step includes parsing a markup language document within the HTTP response (steps 104, 106 of Fig. 4B) and containing the first content object (34 of Figs. 1, 3A and 3B) and the content operation identifier (40 of Figs. 2A and 3B), the content operation identifier including a directive tag (42a of Fig. 2A) specifying the corresponding operation and an object identifier (e.g., 44a, 44b, 44c of Fig. 2A, page 7, lines 21-22) specifying a location of the second content object.

Claim 16 adds to the method of claim 15, wherein the parsing step includes detecting (104 of Fig. 4B) the directive tag as an Hypertext Markup Language (HTML) tag inline prepended (36a of Fig. 3B) to an HTML document (34 of Fig. 3B) specifying the first content object.

Claim 18 adds to the method of claim 14, wherein the parsing step includes parsing the content operation identifier from an HTTP header within the HTTP response (steps 100 and 102 of Fig. 4B, page 10, lines 7-11), the content operation identifier (e.g., 46a of Fig. 2B) including said directive (50 of Fig. 2B) and an object identifier (52 of Fig. 2B, page 8, lines 10-20) specifying a location of the second content object.

Independent claim 20 specifies a server (12b of Fig. 1, page 4, line 27 to page 5, line 2) configured for providing content to a device according to Hypertext Transport Protocol (HTTP). The server comprises an interface (20 of Fig. 1, page 6, lines 10-12 and 22-26) configured for receiving an HTTP request (74 of Fig. 4A) for a first content object (34 of Fig. 1) and outputting an HTTP response (e.g., 60 of Fig. 3B, 84 of Fig. 4A, page 6, lines 22-26, page 9, lines 18-25). The server also includes an executable process (32 of Fig. 1) configured for identifying (70 of Fig. 4A, page 9, lines 6-13) a content operation identifier (tag file 36 stores content operation tags 40 of Fig. 2A or extensible HTTP headers of Fig. 2B, page 7, lines 11-15, page 8, lines 10-13) that identifies a corresponding second content object (44 of Fig. 2A or 52 of Fig. 2B, page 7,

lines 16-18 and 21-24; page 8, lines 11-13 and 16-18, page 10, lines 10-11) determined as relevant to the first content object by a predictive caching operation (70 of Fig. 4A, page 2, lines 5-6 and page 9, lines 6-10), the content operation identifier including a directive (e.g., 42a of Fig. 2A, page 7, lines 16-17; 50 of Fig. 2B, page 8, lines 11-18) for prefetching (page 7, lines 12-24) the second content object as a content operation distinct from presentation of the first content object by the device (page 8, lines 7-9), the executable process configured for supplying (80 or 82 of Fig. 4A, page 9, lines 19-23) within the HTTP response (e.g., 60 of Fig. 3B) the first content object (34 of Fig. 3B) and the content operation identifier (36a of Fig. 3B), enabling the device to perform the prefetching of the second content object based on receipt of the content operation identifier within the HTTP response and distinct from the presentation of the first content object (steps 94 through 106 of Fig. 4B, page 9, line 26 to page 10, line 23).

Claim 21 adds to the server of claim 20, wherein the executable process is configured for retrieving (76 of Fig. 4A, page 9, lines 14-18), based on retrieval of a first stored file (34 of Fig. 1) containing the first content object, a second stored file (36 of Fig 1.) associated with the first stored file and containing the content operation identifier.

Claim 22 adds to the server of claim 21, wherein the executable process is configured for adding to the first content object (34 of Figs. 1, 3A, and 3B) a content operation tag (36a of Fig. 2A includes content operation tag of Fig. 2A, page 7, lines 11-15) that specifies the content operation identifier including a directive tag (42a of Fig. 2A, page 7, lines 16-20) specifying the corresponding content operation to be performed by the device and an object identifier (e.g., 44a, 44b, 44c of Fig. 2A, page 7, lines 21-22) that specifies a location of the second content object.

Claim 23 adds to the server of claim 22, wherein the first content object is a Hypertext Markup Language (HTML) document (34 of Fig. 3A), the executable process configured for inline prepending (82 of Fig. 4A, 60 of Fig. 3B, page 9, lines 19-23) the content operation tag from the second stored file into the HTML document.

Claim 25 adds to the server of claim 21, wherein the executable process is configured for inserting into the HTTP response at least one extensible HTTP header (e.g., 46a of Fig. 2B, step 80 of Fig. 4A, page 8, lines 10-20; page 9, lines 19-20) that specifies the content operation

identifier including said directive (50 of Fig. 2B) to be performed by the device and an object identifier (52 of Fig. 2B) that specifies a location of the second content object.

Claim 27 adds to the server of claim 20, wherein the executable process is configured for adding to the first content object (34 of Figs. 1, 3A, and 3B) a content operation tag (36a of Fig. 2A includes content operation tag 40 of Fig. 2A, page 7, lines 11-15) that specifies the content operation identifier including a directive tag (42a of Fig. 2A, page 7, lines 16-20) specifying the corresponding content operation to be performed by the device and an object identifier (e.g., 44a, 44b, 44c of Fig. 2A, page 7, lines 21-22) that specifies a location of the second content object.

Claim 28 adds to the server of claim 27, wherein the first content object is a Hypertext Markup Language (HTML) document (34 of Fig. 3A), the executable process configured for inline prepending (82 of Fig. 4A, 60 of Fig. 3B, page 9, lines 19-23) the content operation tag into the HTML document.

Claim 30 adds to the server of claim 20, wherein the executable process is configured for inserting into the HTTP response at least one extensible HTTP header (e.g., 46a of Fig. 2B, step 80 of Fig. 4A, page 8, lines 10-20; page 9, lines 19-20) that specifies the content operation identifier including said directive (50 of Fig. 2B) to be performed by the device and an object identifier (52 of Fig. 2B) that specifies a location of the second content object.

Independent claim 32 specifies a proxy device (16a of Fig. 1) configured for retrieving content for a device (16b or 14b of Fig. 1) according to Hypertext Transport Protocol. The proxy device comprises an HTTP interface (20 of Fig. 1, page 6, lines 10-13) configured for sending (92 of Fig. 4B, page 9, line 26 to page 10, line 2) an HTTP request for a first content object (34 of Figs. 1, 3A, 3B), received from the device (14b or 16b of Fig. 1, page 9, line 26 to page 10, line 2), to a destination server (12b of Fig. 1, page 10, line 2) specified by the HTTP request, and receiving from the destination server an HTTP response to the HTTP request (step 94 of Fig. 4B, page 10, lines 2-4) that includes the first content object (34 of Figs. 1, 3A, 3B) and a content operation identifier (40 of Fig. 2A or extensible HTTP headers 46 of Fig. 2B, page 7, lines 11-15, page 8, lines 10-13 and 21-26) that specifies a directive (42a of Fig. 2A, page 7, lines 16-17; 50 of Fig. 2B, page 8, lines 11-18 and 22-26) for prefetching an identified second content object

(page 7, lines 12-24) as an operation to be performed on an identified second content object and distinct from presentation of the first content object. The proxy device also includes an executable resource configured for sending via the HTTP interface the first content object (98 of Fig. 4B, page 10, lines 5-6) to the device (14b or 16b of Fig. 1, page 10, lines 5-6), and executing the operation of prefetching the second content object in response to the content operation identifier (steps 102 or 106 by proxy agent 22 of proxy 16a of Fig. 1, page 10, lines 7-23).

Claim 33 adds to the proxy device of claim 32, wherein the executable resource is configured for parsing the HTTP response (step 96 of Fig. 4B, page 10, lines 2-4) to detect the content operation identifier, the executable resource accessing the identified second content object for execution of the operation (steps 102 or 104 of Fig. 4B, page 10, lines 7-23).

Claim 34 adds to the proxy device of claim 33, wherein the executable resource is configured for parsing a markup language document within the HTTP response (steps 104, 106 of Fig. 4B) and containing the first content object (34 of Figs. 1, 3A and 3B) and the content operation identifier (40 of Figs. 2A and 3B), the content operation identifier including a directive tag (42a of Fig. 2A) specifying the corresponding operation and an object identifier (e.g., 44a, 44b, 44c of Fig. 2A, page 7, lines 21-22) specifying a location of the second content object.

Claim 35 adds to the proxy device of claim 34, wherein the executable resource is configured for detecting (104 of Fig. 4B) the directive tag as an Hypertext Markup Language (HTML) tag inline prepended (36a of Fig. 3B) to an HTML document (34 of Fig. 3B) specifying the first content object.

Claim 37 adds to the proxy device of claim 33, wherein the executable resource is configured for parsing the content operation identifier from an HTTP header within the HTTP response (steps 100 and 102 of Fig. 4B, page 10, lines 7-11), the content operation identifier (e.g., 46a of Fig. 2B) including said directive (50 of Fig. 2B) and an object identifier (52 of Fig. 2B, page 8, lines 10-20) specifying a location of the second content object.

6. Grounds of Rejection to be Reviewed on Appeal:

A. Whether claims 1-2, 13-14, 20-21, and 32-33, are unpatentable under 35 USC §102 in view of U.S. Patent Publication No. 2003/0061451 by Beyda.

B. Whether claims 3-4, 6, 8-9, 11, 15-16, 18, 22-23, 25, 27-28, 30, 34-35, and 37 are unpatentable 35 USC §103 in view of Beyda and Schloss

7. Arguments:

A. **Claims 1, 13, 20, and 32 are not anticipated under 35 U.S.C. §102 in view of Beyda.**

The Examiner finally rejected independent claims 1, 13, 20, and 32 under 35 USC §102 in view of Beyda. Claims 1, 13, 20 and 32 are neither anticipated nor rendered obvious by Beyda for the following reasons.

A1. **Beyda Does Not Disclose or Suggest the Claimed HTTP Response Including the First Content Object and Directive for Prefetching an Identified Second Content Object**

Beyda fails to disclose (expressly or inherently) the claimed feature in independent claims 1, 13, 20, and 32 of sending to a device (or receiving from a destination server) an HTTP response to an HTTP request for a first content object, where the HTTP response includes not only the first content object that was requested in the HTTP request, but also a content operation identifier specifying a ***directive for prefetching a second content object*** as a content operation ***distinct from presentation of the first content object***.

Independent claims 1 and 20 each specify receiving an HTTP request for a first content object, and outputting an HTTP response that includes the first content object (requested in the HTTP request) ***and*** a directive for prefetching a second content object as a content operation distinct from presentation of the first content object ***by the device.*** Independent claim 1 specifies

"sending to the device an HTTP response to the HTTP request"; independent claim 20 specifies "an interface configured for receiving an HTTP request ... and outputting an HTTP response", where the content operation identifier "enabl[es] the device to perform the prefetching of the second content object based on receipt of the content operation identifier within the HTTP response".

Independent claims 13 and 32 each specify "sending an HTTP request for a first content object, received from the device, to a destination server specified by the HTTP request", "receiving from the destination server an HTTP response to the HTTP request that includes the first content object and ... a directive for prefetching an identified second content operation...."

Hence, the claims explicitly specify that the HTTP response, that includes the first content object and the directive for prefetching, is sent ***to the device*** in claim 1, received ***from the destination server*** in claims 13 and 32, and output ***by the interface of the server*** of claim 20. Hence, each of the independent claims inherently require that the HTTP response be transferred between devices according to HTTP protocol (e.g., sent *to the device*, received *from the destination server*, output *by the interface of the server*); consequently, it is insufficient that the HTTP response is generated by a device, but the HTTP response also must be transferred "to the device", "from the destination server", or output by an interface of the server.

Hence, the claims specify that the HTTP response that is transferred to (or from) a device (or destination server) includes not only the first content object that was requested in order to enable a requesting device to present the requested first content object, but the HTTP response also includes a directive for prefetching the second content object: the claims also specify that the directive enables prefetching of the second content object as a content operation distinct from presentation of the first content object.

The Examiner has the burden of establishing that Beyda discloses each and every element of the claim such that the identical invention must be shown in as complete detail as is contained

in the claim.[1] Further, Further, anticipation cannot be established based on a piecemeal application of the reference, where the Examiner picks and chooses isolated features of the reference in an attempt to synthesize the claimed invention.[2] In other words, it is not sufficient that a single prior art reference discloses each element that is claimed, but the reference also must disclose that the elements are arranged as in the claims under review. *In re Bond*, 15 USPQ2d 1566, 1567 (Fed. Cir. 1990) (citing *Lindemann Maschinenfabrik GmbH*).

In other words, the Examiner has the burden of establishing not only that Beyda discloses an HTTP response output by a server, or that a proxy device may perform caching of requested content, but that an HTTP response is output from a server (or received by a proxy) ***in the same manner as claimed***, namely that the HTTP response output *to the device* (claims 1, 20), or received *from the destination server* (claims 13, 32), includes both the requested first content object ***and the directive for prefetching the identified second content object***.

Beyda describes with respect to Figure 1 a local server 14 having a proxy server 16 "where caching is accomplished" (para. 16, lines 5-6): the proxy server 16 includes a web cache and a table, illustrated in Figure 2, that "keeps track of URLs of all web pages that are requested by any of the clients 10, 12, and 14." (Paragraph 17, lines 3-4). For each URL listed, Beyda describes that "the table keeps the time the last client accessed the webpage, and the corresponding modification timestamp of when the page was last modified." (Para. 17, lines

---

[1] As specified in MPEP §2131: "'A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference' *Verdegaal Bros. V. Union Oil Co. of California,* 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). ... 'The identical invention must be shown in as complete detail as is contained in the ... claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989)." MPEP 2131 (Rev. 3, Aug. 2005, at p. 2100-76).

[2] "Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim." *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 221 USPQ 481, 485 (Fed. Cir. 1984). "Anticipation cannot be predicated on teachings in the reference which are vague or based on conjecture." *Studiengesellschaft Kohle mbH v. Dart Industries, Inc.*, 549 F. Supp. 716, 216 USPQ 381 (D. Del. 1982), *aff'd.*, 726 F.2d 724, 220 USPQ 841 (Fed. Cir. 1984).

5-7). "When a client sends a request to the local server 14 for a web page, the URL of the requested web page is searched in the table [of] Figure 2. If the requested URL is not found, [] the local server 14 directs [i.e., forwards] the request to the remote server 18 via the Internet/Intranet 20." (Para. 19, lines 1-5). The retrieved web page is cached in the proxy server 16, and the associated details regarding the URL, time, and timestamp are captured in the table of figure 2 (Para. 19).

If the URL is found in the table of Fig. 2, the *local server* 14 sends the request to the remote server 18: if the time stamp of the web page from the remote server 18 matches the time stamp in the table, the *local server* 14 stops the transfer and delivers the cached content in the proxy server to the client (Para. 20, lines 1-7).

Paragraphs 21-28 describe a predicted prefetch of web pages by the *local server* 14: the local server 14 keeps track of the time-based pattern of requested web pages, and divides the usage pattern into a certain predetermined time to record the hit rate of every webpage visited, and rank the webpages according to hit rate. (Para. 21). The usage pattern is analyzed for repeating patterns (Para. 22-23), and the determined patterns are used to predictably prefetch webpages by the local server 14 into a cache (Para. 24-27).

Hence, the *local server* 14 performs predictive prefetching of web page content based on determining repeating usage patterns, in order to limit or reduce the required access via the Internet to elements having encountered a change from the last access time and the most recent web page (Para. 27).

The rejection fails to establish that the cited reference discloses or suggests the claimed ***HTTP response*** that includes ***both the first content object*** (for presentation of the first content object by the device), ***and*** the content operation identifier that includes ***a directive for prefetching the second content object as a content operation distinct from presentation of the first content object***. In fact, the rejection cites Para. 21-28 of Beyda, but fails to specifically identify *any* feature in Beyda within the cited portion that can be considered a disclosure of the claimed directive for prefetching, as claimed. As demonstrated above, Beyda simply describes a local server 14 performing a predictive caching operation to fetch content from a remote server

18, and that locally caches the content for subsequent use during high-traffic intervals by clients connected to the local server 14. The rejection fails to demonstrate that Beyda discloses or suggests that ***a single HTTP response*** includes ***both*** the first content object ***and*** the content operation identifier (including the directive for prefetching), as claimed.

For these reasons alone, the §102 rejection should be withdrawn.

The Advisory Action mailed April 3, 2007 further demonstrates the deficiencies in the §102 rejection. For example, the Advisory Action parrots the initial rejection and claim language to assert the following:

Beyda suggests HTTP response including both the first content object and the content operation identifier that includes a directive for prefetching the second content object as a content operation distinct from presentation of the first content object [see figs. 1-2 and paragraphs 0017-0028].

However, the Advisory Action provides no more support for above assertion than the following statement:

"For example, Beyda suggests HTTP response [paragraphs 0016-0028]."

Hence, the rejection and Advisory Action fail to demonstrate that Beyda discloses that the HTTP response includes not only the first content object (having been requested in an HTTP request), but ***further includes*** a ***directive for prefetching a second content object*** as a content operation ***distinct from presentation of the first content object***.

A2. The Rejection Improperly Ignores Explicit Claim Language that the Directive for Prefetching is within an HTTP Response Output to/Received by Another Device Distinct from the Originator of the Directive

The Advisory Action also provides the specious argument that "the claims specify a directive for refetching [sic] which is nothing more than pre-ftetching [sic] portion of a single cache e.g. its URL [see paragraphs 0017-0028]." This argument, however, demonstrates an improper disregard of explicit claim language that specify ***a single HTTP response*** is sent to a device (claims 1 and 20), or received from a destination server (claims 13 and 32) that is *distinct*

from the device having generated the HTTP response: that ***single HTTP response*** includes both the first content object (requested by a device) and the ***directive for prefetching a second content object***; further, the directive for prefetching enables any device receiving the HTTP response to prefetch the second content object distinct from presentation of the first content object. It is well settled that all words in the claim must be considered.[3]

Beyda provides no disclosure that the HTTP response from the remote server 18 includes anything other than the requested content (i.e., the first content object). In fact, Beyda teaches away from the claims by requiring the **local server 14** to perform the predictive prefetching based on its own analysis of its cached data (see para. 21-24). Beyda also provides no disclosure that any HTTP response output by the local server 14 to the requesting client devices 10, 12, or 14 includes anything other than the requested content (i.e., the first content object); to the contrary, Beyda teaches no more than the local server 14 delivering content cached within the local server 14 to the client devices 10, 12, or 14, or passing updated content received from the remote server 18 to the client devices 10, 12, or 14 (see, e.g., para. 19-20 and 27).

Hence, although Beyda discloses a local server 14 that performs its own predictive prefetching based on reviewing its cache (illustrated in Figure 2), Beyda provides no disclosure or suggestion that the predictive prefetching is added within an ***HTTP response***, enabling another device receiving the HTTP response to perform the prefetching of the second content object in response to receipt of the content operation identifier specifying the directive for prefetching.

Further, the rejection fails to present any interpretation that should extend beyond the broadest reasonable interpretation of the claimed "directive for prefetching": the broadest *reasonable* interpretation must be (1) consistent with the specification, and (2) consistent with

---

[3]"All words in a claim must be considered in judging the patentability of that claim against the prior art." *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970)."

the interpretation that those skilled in the art would reach.[4] The specification explicitly describes prefetching as fetching new content without relying on a client request to provide content acceleration (see, e.g., Title, page 10, line 12 to page 11, line 2) of the second content object, and each of the claims specify that the second content object that is to be prefetched is explicitly identified in the HTTP response, and is distinct from ***presentation of the first content object by the device***.

Further, use of the term "prefetching" is notoriously well known in the art as fetching *prior to being requested*.

Hence, the claims explicitly specify that the HTTP response, that includes both the first content object and the directive for prefetching, is sent ***to the device*** in claim 1, received ***from the destination server*** in claims 13 and 32, and output ***by the interface of the server*** of claim 20.

Hence, the deliberate disregard of the claimed feature of a single HTTP response including both the first content object (for presentation of the first content object) and the directive for prefetching the second content object distinct from the presentation of the first content object is reversible error because "[a]ll words in a claim must be considered in judging the patentability of that claim against the prior art." *In re Wilson*, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970).

For these and other reasons, the §102 rejection of independent claims 1, 13, 20, and 32 should be withdrawn.

B. Claims 3-4, 6, 8-9, 11, 15-16, 18, 22-23, 25, 27-28, 30, 34-35, and 37 are Not Rendered Obvious Under 35 USC §103 In View of Beyda and Schloss

---

[4]"During patent examination, the pending claims must be 'given their broadest reasonable interpretation consistent with the specification.'" MPEP §2111 at 2100-46 (Rev. 3, Aug. 2005) (*quoting In re Hyatt*, 211 F.3d 1367, 1372, 54 USPQ2d 1664, 1667 (Fed. Cir. 2000)).

"The broadest reasonable interpretation of the claims must also be consistent with the interpretation that those skilled in the art would reach." MPEP §2111.01 at 2100-47 (Rev. 3, Aug. 2005) (*citing In re Cortright*, 165 F.3d 1353, 1359, 49 USPQ2d 1464, 1468 (Fed. Cir. 1999)).

The §103 rejection of claims 3-4, 6, 8-9, 11, 15-16, 18, 22-23, 25, 27-28, 30, 34-35, and 37 improper because it fails to demonstrate that "there was an apparent reason to combine the known elements *in the fashion claimed* by the [claims] at issue [where] this *analysis should be made explicit.*" *KSR Int'l v. Teleflex, Inc.* No. 04-1350, Slip. op. at 14, 82 USPQ2d 1385, 1396 (U.S. Apr. 30, 2007).

The Examiner has the burden of demonstrating that "there was an apparent reason to combine the known elements *in the fashion claimed.*" *KSR Int'l v. Teleflex, Inc.* No. 04-1350, Slip. op. at 14, 82 USPQ2d 1385, 1396. The Examiner has failed to establish the analysis as required by the Supreme Court. Rather, the hypothetical combination teaches no more than "the predictable use of prior art elements according to their established functions," *Id.*, with no disclosure or suggestion of the claimed features as a whole.

Claims 3, 6, 8, 11, 15, 22, 25, 27, 30, 34, and 37 each specify that the content operation identifier in the HTTP response contains a ***directive tag*** (or an extensible HTTP header) specifying the corresponding operation and an object identifier specifying a location of the second content object. In addition, claims 3, 8, 22 and 27 specify the specific operation of ***adding*** to the first content object a content operation tag that specifies the content operation identifier including the directive tag.

As admitted in the Final Action, Beyda does not disclose "adding to the first content object a content operation tag that specifies the content operation identifier". As described below, however, (1) Schloss does not disclose or suggest the claimed ***adding*** the content operation tag to the existing first content object, but rather replaces noncacheable content with identifiers; (2) further, Schloss replaces the noncacheable content with an identifier, and does not disclose or suggest adding a ***directive*** (or an extensible HTTP header), as claimed.

Schloss discloses replacing portions of a web page (persistent object fragments) with identifiers in the web page in order to improve the caching ability of the web page. In particular, Schloss describes that typically "a document is not cached even if only a small fraction of its content is dynamic" (col. 2, lines 21-22); hence, Schloss describes a system that parses a web

object (i.e., a web page) to identify "persistent object fragments" (e.g., dynamic objects or large objects deemed uncacheable), and replace the persistent object fragments with "persistent object fragment *identifiers*" that render the modified web page more cacheable at the client device. (See, e.g., Figs. 2-8, col. 4, lines 39-54; col. 5, line 36 to col. 6, line 46; col. 7, lines 7-38; and col. 9, lines 7-55).

In particular, Schloss specifies at col. 9, lines 49-55:

> According to the present invention, the server uses persistent object fragment identifiers to replace persistent object fragments (such as dynamic objects or large segments) in a Web object. The revised object is thus more cacheable at the client device, since the server has removed the dynamic or large objects from the object and reduced the size of the object.

Although Schloss improves caching of web content, Schloss provides no disclosure whatsoever of ***prefetching content***, as claimed. Rather, Schloss modifies a web page to make the web page more cacheable, and sends the modified web page having improved cacheability to a destination device.

Further, there is no disclosure or suggestion in Schloss of ***adding*** a content *operation* tag; rather Schloss simply replaces uncacheable objects with "persistent object fragment *identifiers*" that render the modified web page more cacheable at the client device.

Finally, there is no reference whatsoever to an extensible HTTP header, as specified in claims 7, 11, 25, and 30.

The rejection provides an argument why one skilled in the art would have combined the teachings of Beyda and Schloss *generally* (i.e., according to their predictable use); however, the rejection fails to provide any analysis of any "apparent reason" that one of ordinary skill in the art would have provided any improvements *beyond* (i.e., more than) the predictable use of Beyda and Schloss according to their established functions.[5]

Assuming one skilled in the art would modify Beyda with Schloss, this hypothetical combination still would neither disclose nor suggest the claimed ***adding*** in claims 3, 8, 22, and

---

[5] See *KSR Int'l v. Teleflex, Inc.* No. 04-1350, Slip. op. at 13-14, 82 USPQ2d 1385, 1396.

27 to the first content object a ***directive tag***, let alone receiving an HTTP response that includes the first content object ***and the directive tag***, as specified in claims 3, 8, 15, 22, 27, and 34, or an HTTP response that includes an extensible HTTP header as specified in claims 7, 11, 25, and 30. Rather, the hypothetical combination simply would teach replacing uncacheable objects with **identifiers**.

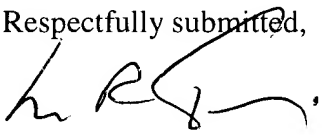
Hence, the §103 rejection fails to demonstrate that it would have been obvious to arrive at the claimed combinations. "[A] patent composed of several elements is not proved obvious merely by demonstrating that each of its elements was, independently, known in the prior art." *KSR Int'l v. Teleflex, Inc.*, Slip op. at 13, 82 USPQ 2d 1385, 1396 (U.S. Apr. 30, 2007).

Conclusion

For the reasons set forth above, it is clear that Appellant's claims 1-4, 6, 8-9, 11, 13-16, 18, 20-23, 25, 27, 28, 30, 32-35 and 37 are patentable over the references applied. Accordingly the appealed claims 1-4, 6, 8-9, 11, 13-16, 18, 20-23, 25, 27, 28, 30, 32-35 and 37 should be deemed patentable over the applied references. It is respectfully requested that this appeal be granted and that the Examiner's rejections be reversed.

To the extent necessary, Appellant petitions for an extension of time under 37 C.F.R. 1.136 and 37 C.F.R. 41.37(e). Please charge any shortage in fees due in connection with the filing of this paper, including any missing or insufficient fees under 37 C.F.R. 1.17(a) or 41.20(b)(2), to Deposit Account No. 50-1130, under Order No. 95-472, and please credit any excess fees to such deposit account.

Respectfully submitted,



Leon R. Turkevich
Registration No. 34,035

Customer No. 23164
September 10, 2007

8. APPENDIX – CLAIMS ON APPEAL

1. (PREVIOUSLY PRESENTED) A method of providing content to a device according to Hypertext Transport Protocol (HTTP), the method comprising:

receiving an HTTP request for a first content object;

identifying a content operation identifier that identifies a corresponding second content object determined as relevant to the first content object by a predictive caching operation, the content operation identifier including a directive for prefetching the second content object as a content operation distinct from presentation of the first content object by the device; and

sending to the device an HTTP response to the HTTP request, the HTTP response including the first content object and the content operation identifier, enabling the device to perform the prefetching of the second content object based on receipt of the content operation identifier and distinct from the presentation of the first content object.

2. (ORIGINAL) The method of claim 1, wherein the identifying step includes retrieving, based on retrieval of a first stored file containing the first content object, a second stored file associated with the first stored file and containing the content operation identifier.

3. (ORIGINAL) The method of claim 2, wherein the sending step includes adding to the first content object a content operation tag that specifies the content operation identifier including a directive tag specifying the corresponding content operation to be performed by the device and an object identifier that specifies a location of the second content object.

4. (ORIGINAL) The method of claim 3, wherein the first content object is a Hypertext Markup Language (HTML) document, the adding step including inline prepending the content operation tag from the second stored file into the HTML document.

6. (PREVIOUSLY PRESENTED) The method of claim 2, wherein the sending step

includes inserting into the HTTP response at least one extensible HTTP header that specifies the content operation identifier including said directive to be performed by the device and an object identifier that specifies a location of the second content object.

8. (ORIGINAL) The method of claim 1, wherein the sending step includes adding to the first content object a content operation tag that specifies the content operation identifier including a directive tag specifying the corresponding content operation to be performed by the device and an object identifier that specifies a location of the second content object.

9. (ORIGINAL) The method of claim 8, wherein the first content object is a Hypertext Markup Language (HTML) document, the adding step including inline prepending the content operation tag into the HTML document.

11. (PREVIOUSLY PRESENTED) The method of claim 1, wherein the sending step includes inserting into the HTTP response at least one extensible HTTP header that specifies the content operation identifier including the directive to be performed by the device and an object identifier that specifies a location of the second content object.

13. (PREVIOUSLY PRESENTED) A method of retrieving content for a device according to Hypertext Transport Protocol, the method comprising:

first sending an HTTP request for a first content object, received from the device, to a destination server specified by the HTTP request;

receiving from the destination server an HTTP response to the HTTP request that includes the first content object and a content operation identifier that specifies a directive for prefetching an identified second content object as an operation to be performed on the identified second content object and distinct from presentation of the first content object;

second sending the first content object to the device; and

executing the operation of prefetching the second content object in response to the

content operation identifier.

14. (ORIGINAL) The method of claim 13, wherein the executing step includes:

detecting the content operation identifier based on parsing the HTTP response; and

accessing the identified second content object for execution of the operation.

15. (ORIGINAL) The method of claim 14, wherein the detecting step includes parsing a markup language document within the HTTP response and containing the first content object and the content operation identifier, the content operation identifier including a directive tag specifying the corresponding operation and an object identifier specifying a location of the second content object.

16. (ORIGINAL) The method of claim 15, wherein the parsing step includes detecting the directive tag as an Hypertext Markup Language (HTML) tag inline prepended to an HTML document specifying the first content object.

18. (PREVIOUSLY PRESENTED) The method of claim 14, wherein the parsing step includes parsing the content operation identifier from an HTTP header within the HTTP response, the content operation identifier including said directive and an object identifier specifying a location of the second content object.

20. (PREVIOUSLY PRESENTED) A server configured for providing content to a device according to Hypertext Transport Protocol (HTTP), the server comprising:

an interface configured for receiving an HTTP request for a first content object and outputting an HTTP response; and

an executable process configured for identifying a content operation identifier that identifies a corresponding second content object determined as relevant to the first content object

by a predictive caching operation, the content operation identifier including a directive for prefetching the second content object as a content operation distinct from presentation of the first content object by the device, the executable process configured for supplying within the HTTP response the first content object and the content operation identifier, enabling the device to perform the prefetching of the second content object based on receipt of the content operation identifier within the HTTP response and distinct from the presentation of the first content object.

21. (ORIGINAL) The server of claim 20, wherein the executable process is configured for retrieving, based on retrieval of a first stored file containing the first content object, a second stored file associated with the first stored file and containing the content operation identifier.

22. (ORIGINAL) The server of claim 21, wherein the executable process is configured for adding to the first content object a content operation tag that specifies the content operation identifier including a directive tag specifying the corresponding content operation to be performed by the device and an object identifier that specifies a location of the second content object.

23. (ORIGINAL) The server of claim 22, wherein the first content object is a Hypertext Markup Language (HTML) document, the executable process configured for inline prepending the content operation tag from the second stored file into the HTML document.

25. (PREVIOUSLY PRESENTED) The server of claim 21, wherein the executable process is configured for inserting into the HTTP response at least one extensible HTTP header that specifies the content operation identifier including said directive to be performed by the device and an object identifier that specifies a location of the second content object.

27. (ORIGINAL) The server of claim 20, wherein the executable process is configured for adding to the first content object a content operation tag that specifies the content operation

identifier including a directive tag specifying the corresponding content operation to be performed by the device and an object identifier that specifies a location of the second content object.

28. (ORIGINAL) The server of claim 27, wherein the first content object is a Hypertext Markup Language (HTML) document, the executable process configured for inline prepending the content operation tag into the HTML document.

30. (PREVIOUSLY PRESENTED) The server of claim 20, wherein the executable process is configured for inserting into the HTTP response at least one extensible HTTP header that specifies the content operation identifier including said directive to be performed by the device and an object identifier that specifies a location of the second content object.

32. (PREVIOUSLY PRESENTED) A proxy device configured for retrieving content for a device according to Hypertext Transport Protocol, the proxy device comprising:

an HTTP interface configured for sending an HTTP request for a first content object, received from the device, to a destination server specified by the HTTP request, and receiving from the destination server an HTTP response to the HTTP request that includes the first content object and a content operation identifier that specifies a directive for prefetching an identified second content object as an operation to be performed on an identified second content object and distinct from presentation of the first content object; and

an executable resource configured for sending via the HTTP interface the first content object to the device, and executing the operation of prefetching the second content object in response to the content operation identifier.

33. (ORIGINAL) The proxy device of claim 32, wherein the executable resource is configured for parsing the HTTP response to detect the content operation identifier, the executable resource accessing the identified second content object for execution of the operation.

34. (ORIGINAL) The proxy device of claim 33, wherein the executable resource is configured for parsing a markup language document within the HTTP response and containing the first content object and the content operation identifier, the content operation identifier including a directive tag specifying the corresponding operation and an object identifier specifying a location of the second content object.

35. (ORIGINAL) The proxy device of claim 34, wherein the executable resource is configured for detecting the directive tag as an Hypertext Markup Language (HTML) tag inline prepended to an HTML document specifying the first content object.

37. (PREVIOUSLY PRESENTED) The proxy device of claim 33, wherein the executable resource is configured for parsing the content operation identifier from an HTTP header within the HTTP response, the content operation identifier including said directive and an object identifier specifying a location of the second content object.

9. Evidence Appendix

[No evidence attached]

10. Related Proceedings Appendix

[No Related Proceedings]